

Aalto University  
School of Science  
Master's Programme in ICT Innovation

Denis Laure

## **Bringing Understanding of Simulation Material to Interaction Designers**

Master's Thesis

Espoo, August 4, 2016

Supervisor: Professor Marko Nieminen, Aalto University

Instructor: Bogdan Timus, PhD

Aalto University School of Science Degree Programme in Computer Science and Engineering Master's Programme in ICT Innovation		ABSTRACT OF THE MASTER'S THESIS	
Author: Denis Laure			
Title: Bringing Understanding of Simulation Material to Interaction Designers			
Number of pages: 37	Date: 04.08.2016	Language: English	
Major: Human Computer Interaction and Design			
Supervisor: Marko Nieminen			
Advisor: Bogdan Timus			
<p>Abstract:</p> <p>Simulation-based tools are complex and obscure software. However, a company may benefit from using such tools, as they provide more precise and accurate information. Ericsson has developed a RAN simulator that allows to model cellular networks taking into account even their smallest aspects. This allows to compare different solutions for particular case and select the best one. Therefore, Ericsson can propose better, i. e. most efficient and less cost, solutions to its customers. However, the RAN simulator is developed in MATLAB and does not have any graphical user interface. Therefore, it is not possible for people who manage sales at Ericsson to use it, because they have no skills for it. This, in fact, raises the need of development of a tool that will provide sales people with a convenient way to access the RAN simulator.</p> <p>This research describes a process of prototyping three simulation-based tools for Ericsson. It covers a process of providing interaction designers with the knowledge about simulations. The research gives insights on important details of the simulations that are needed to be delivered to the designers, as well as aspects of developing simulation-based tools within multidisciplinary team. Moreover, the research introduces a “mediator person” who can significantly help and improve the process of the development of simulation-based tools.</p>			
Keywords: simulations, digital material, interaction design.			

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Theory and Related Research</b>	<b>4</b>
<b>3</b>	<b>Method</b>	<b>7</b>
<b>4</b>	<b>Results</b>	<b>9</b>
4.1	Solution Architecture . . . . .	9
4.2	Carrier Aggregation Tool . . . . .	11
4.2.1	Design Challenges . . . . .	16
4.2.2	Technical Challenges . . . . .	17
4.3	Raw RAN Simulations Tool . . . . .	18
4.4	Administration Tool . . . . .	20
<b>5</b>	<b>Discussion</b>	<b>25</b>
<b>6</b>	<b>Conclusions</b>	<b>29</b>
<b>7</b>	<b>Acknowledgments</b>	<b>30</b>
<b>A</b>	<b>Back-end Part Description</b>	<b>35</b>

# 1 Introduction

Ericsson is one of the world's leading companies in the area of communications technology. Ericsson provides information and communication technologies infrastructure as well as software and services in this field. It is a big technology driven multinational corporation that states quality and top edge technologies as one of its most important values [1].

Ericsson is a big company with over 115 000 employees all over the world [2]. Therefore, it has a complex organizational structure with multiple different units and departments having their own aims and responsibilities within the company.

One such department is the Research and Development (R&D) department, whose main task is to study new technologies, analyze future potential challenges and propose new solutions that Ericsson might develop and offer in the future. Currently, one of the many projects that the R&D department is working on is Radio Access Network (RAN) simulator. It is written in MATLAB and provides the ability to simulate complex detailed cases for cellular network prediction purposes. It takes into account even subtle details, such as city environment and buildings' details and due to this fact can calculate precise characteristics of the network for particular case, allowing to choose the best solutions.

The RAN simulator is written in MATLAB and, therefore, is aimed to be used by R&D employees who are very experienced in working with MATLAB as well as in radio physics. Therefore, their work is usually guided by functionality and not usability. However, nowadays accessibility and usability play an important role in system development. Modern software needs to be accessible by anyone, from anywhere and at any time [3]. Therefore, human-computer interaction experts are needed within a team that is aimed to develop a high quality modern system. Moreover, it is important for them to communicate and cooperate with software engineers and domain experts in order for their work to be successful and productive [4].

Another Ericsson department, the Sales Support department, manages the sales of cellular network solutions all over the world and also helps sales people in the regions. People in this department could potentially benefit from simulation tools, such as the RAN simulator, even though it was initially developed for internal use of the R&D department. Based on the

RAN simulations the Sales Support department could provide more qualified service by making precise pre-sales calculations that would possibly decrease the cost of prospective end solutions. Therefore, the usage of simulation tools can possibly increase Ericsson's sales by providing competitive advantage.

One example of the cases where the usage of simulation tools can decrease costs of prospective solutions is indoor networks or Small Cells [5]. Small Cells is a modern technology that allows to propagate mobile networks on small areas, substantially inside buildings. Small Cells are much more energy efficient and less costly comparing to macro cells and at the same time they have a number of advantages compared to traditional indoor Wi-Fi networks [6, 7]. The RAN simulator uses complex models that capture aspects which are not covered by any of the currently available commercial products. By leveraging on this analysis capability, Ericsson can help its customers to make better use of their investment budgets.

Unfortunately, the RAN simulator is written in MATLAB and is distributed as a set of MATLAB functions and scripts and, thus, does not have any Graphical User Interface. Therefore, it cannot be used by managers at the Sales Support department, which draws a clear need of development of a software that will provide Sales Support people an easy-to-use and convenient way to use simulations in their work.

## 2 Theory and Related Research

Development of simulation-based software cannot be done without having any knowledge of the underlying technologies and software, which in this case is the RAN simulator. The usual way of designing software is that the designer creates an interface relying only on the user's needs and user research, passing then the outcomes to developers [8] who treat it as a specifications. Therefore, the designer loses the important part of design process — *conversation with immaterial materials* [9]. This in turn leads to missing the knowledge about the material, its properties and possibilities, which may lead to linear and one-directional view at the problem and bordering of the view point for creating ideas and solutions [10].

The design way outside of the material approach described above was common in the past, when interaction designers mostly worked on projects for desktop computations and, therefore, it was assumed that they know the limits and borders of technologies. In the modern world of novel technologies such approach results into putting too much effort into conceptual parts of a design disregarding material and, in fact, ending up with “fighting the materials to fit with the conceptual idea” [11]. Moreover, it can result in the designer's “dreams” remaining only “dreams” because of not understanding and realizing meanings of the technologies and, therefore, unable to implement these “dreams” because of the technologies limitations [12].

To be able to work with novel and complicated technologies new ways of design are needed. One of such ways might be derived from practices of architects and industrial designers whose design process consists of building artefacts with materials and testing, discussing and criticizing them within the team [13]. However, unlike these fields, the material of interaction designers is *digital*, i.e. a technology that helps to prop up interaction. Digital material can be also assumed similar to what Vallgård and Redström named *computational composites* — a solid combination of hardware, software and traditional materials that allows to introduce new or encase existing properties to architects and industrial designers [14] — but in the interaction design world. However, digital material describes a broad set of different things varying from computers and consumer electronics themselves to specific sensors, electronic parts, software and even programming languages as fundamentals [13].

One example of using digital materials in design process is provided by

Sundström and Höök in [13]. They observed sensor network as a digital material for their FriendSense system that allows friends to express their feelings and experience emotional closeness through movement. Allowing digital material to guide them through the design process resulted into the understanding of limitations of sensors and sensor networks, as well as possibilities of software and algorithms. This knowledge helped them to not step away to the technologically impossible decisions in their design process and, therefore, saved time and allowed to finish with working prototype of the system.

In [9] authors describe two workshops that they conducted with interaction designers to investigate the lack of conversation with immaterial material in the GUI design process. The conclusions from these workshops were that the fact that designers skip material part of the design process emerges into lack of flexibility of ways to transfer the results of the process to developers. The workshops also allowed to clarify a need for a tool that would provide support in refinement of the design and communication between designers, developers and users.

However, digital material sometimes can be too complicated for designers. For instance, knowledge about solid technical fields, such as wireless communications, where even researchers sometimes use wrong models for simulations and make simplistic assumptions [15]. Moreover, unlike traditional materials, digital materials are changing in time, they change and reveal their properties in interaction process. Therefore, it is important to put them in use instead of experiencing them in particular moment to reveal the most potential out of their qualities and properties [16].

Sundström et al. in their paper [10] proposed the *inspirational bits* approach as a way to handle complexity of digital materials. The aim of the approach is to provide understanding and, therefore, allow to start conversation with material for everybody in a multidisciplinary team. Inspirational bits represent simple, quickly made and dirty design objects that allow to expose particular properties of complicated digital material. Usually, inspirational bits also represent an instrument that allows everybody in a team to “play” with material and create quick and dirty low-fidelity prototypes. It allows all members of the team to quickly test their assumptions about materials or check the possibility of design ideas. One example of inspirational bits is described in [10] and is named RadioSound. It consists of two sensor nodes wirelessly connected to each

other. One of the nodes has a speaker that constantly produces the single tone sound. The pitch of this sound tone depends on the signal strength between two nodes. RadioSound allows to “materialize” signal strength property of the radio and, therefore, explore it in different environments.

On the other hand, prototyping can also be assumed as a way to provide knowledge about digital material within the multidisciplinary team [17]. For example, Sundström and Höök in [13] used prototypes to become acquainted with digital materials (sensors and sensor networks) and reveal their properties, qualities and limitations.

However, due to broad meaning of the digital material itself and its complexity, every team has to establish their own way of understanding materials. In this project the RAN simulator and simulations in general are conceptualized as digital material, which properties and limitations are needed to be delivered to interaction designers who work on designing of new tools for Ericsson’s Sales Support department. Therefore, research question that is being answered in this study is:

*What are the aspects of simulation material that are important for interaction designers?*

Moreover, the research also aims to provide insights and ways to improve the work of multidisciplinary team on simulation-based tools.



### 3 Method

The main method of the study is online-prototyping. The method allowed to:

- start conversation with digital material, i.e. get initial knowledge about simulations in general and the RAN simulator in particular, bringing understanding of how it can be used;
- bring understanding of digital material, its properties and limitations to interaction designers through a practical approach;
- evaluate different initial ideas;

The main approach was to quickly develop small prototypes. Each prototype was serving a narrow particular need of a small group of people at the Sales Support department and, therefore, used particular type of simulations. The prototypes were vertical, meaning that each of them implemented only one particular feature that functioned well, as if it was final product. It allowed to evaluate initial ideas and hypotheses regarding the simulation material. Moreover, these prototypes were also task-oriented, i.e. implementing only the features that are necessary to accomplish particular tasks [17]. The choice of this type of prototypes also resulted in the following benefits.

- Our team was able to quickly deliver tools that served the needs of employees. This, in fact, raised attitude towards our team among Ericsson employees with whom we were working together and also gave them practical understanding of our job and capabilities.
- Prototypes allowed to develop the idea and the vision of the whole system that will allow to develop simulation-based tools in future. It also allowed to develop a prototype of this system by combining together vertical prototypes and looking for commonalities among them.

The prototypes were done in close cooperation with interaction designers as well as users. It allowed to always keep conversation about the prototypes and reflect on them while working. The needs of the designers and restrictions of the simulation material were formulated as development

tasks and recorded in the internal project management tool. The tool allowed all the members of the team to track the status of the prototypes development. Additionally, the reflections were also discussed within a team in informal talks. Consequentially, these reflections formed into the final ideas that will be described in Discussion and Conclusions sections of the thesis.

## 4 Results

The project was done within a team together with two other master students who took a role of interaction designers. The project itself was initiated, guided and helped by our Ericsson supervisor, who previously worked at R&D department and has significant knowledge and experience with the RAN simulator and now works at the Sales Support department.

The RAN simulator provides a wide range of different simulations with a big number of parameters. For the Sales Support department employees this means extremely high level of complexity and high entry barrier to start using simulations in their everyday tasks. In most of the cases they just need certain types of simulation with only few parameters. However, these cases depend on particular situations and their number may vary in a very broad range. Therefore, it is impossible to develop one tool that will provide convenient access for the Sales Support department employees to simulations that at the same time will cover all the possible cases.

### 4.1 Solution Architecture

Consequently, it was decided that the architecture of the solution to be delivered should be separated into two parts: front-end and back-end.

The back-end part is implemented as a remote web server that can launch certain type of simulations with particular parameters by remote request. It provides standardized unified RESTful API [18] for launching simulations, checking their status and getting the results. It also provides some common functionality:

- restricted access to certain types of simulations and services that can be granted only by system administrator,
- concurrent asynchronous execution of simulations as separate computational processes,
- store and provide access to information about all the simulations that were performed or currently running on the server,
- store and provide access to different simulation files including result and logging files.

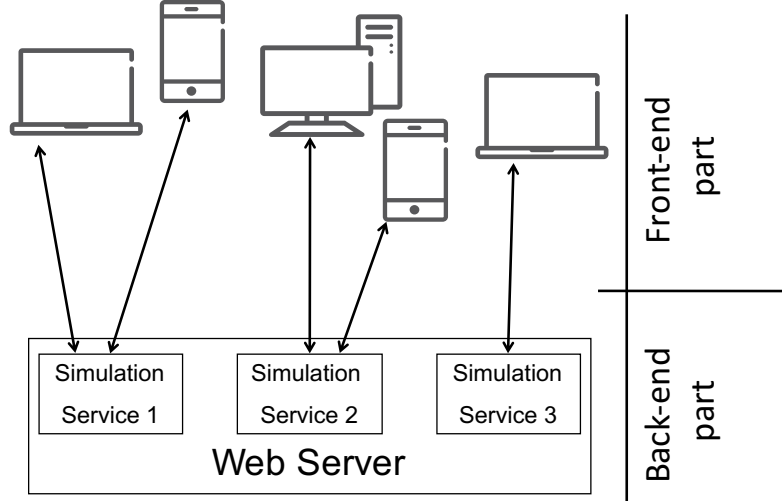


Figure 1: Architecture of the solution.

A detailed technical description of the back-end part implementation is provided in Appendix A.

The front-end part represented as a number of small task-oriented thin clients. The purpose of these small clients is to provide a simple and easy-to-use graphical interface to the back-end server for the end-users of the tool. Therefore, the implementation of front-end clients depends on the particular task they are aimed to solve. They can be either web applications, mobile applications, desktop applications or anything else. Furthermore, it is possible to develop several clients with the same purpose but for different platforms.

The scheme of the architecture is shown on Figure 1.

This kind of architecture allows to separate complex management of simulations from user interface. Therefore, the development of new tools is simplified as these two parts can be developed separately not depending much on each other. Moreover, it allows to quickly change one of the parts according to the changed needs or workflow without affecting the other one.

## 4.2 Carrier Aggregation Tool

The first prototype that was developed for this project was a Carrier Aggregation Tool. The need for the tool was brought by two Sales Support department managers, who were giving a presentation to one of the Ericsson's customers about the advantages of the carrier aggregation technique. Carrier aggregation is a technique that is used in LTE-Advanced standard to significantly increase both uplink and downlink data rates in the network and, therefore, to improve transmission performance [19, 20]<sup>1</sup>. The presentation included several graphs that compared data rates of different network frequency bands with and without use of carrier aggregation for the particular city case. These graphs were produced with the use of the RAN simulator by our team's supervisor.

The presentation was successful and managers decided that they might need to create the same type of presentations for other cases in future. Therefore, it was decided to develop a tool that would automate this process and allow to automatically create the same type of graphs for predefined cases with the use of simulations. Since the main purpose of the tool is to build graphs, it was also decided that the tool itself should be web-based, i.e. its client (or front-end) part should be run in a web browser. This choice of platform provided the following benefits:

- the tool is easy to access for its end users, because there is no need to install any software, but just open a particular web page in web browser;
- all of the team members already had experience in the development of web applications;
- the web application development field is very advanced nowadays and there are a lot of frameworks and libraries that allow to easily include some specific functionalities, such as building complicated graphs, into applications.

The development of the tool was separated as follows. I was responsible for the back-end part of the tool that takes the needed simulation parameters from front-end part, manages launching of the simulations and

---

<sup>1</sup>Since the radio topic is not the aim of the paper, I will not bother the reader describing this technique more.

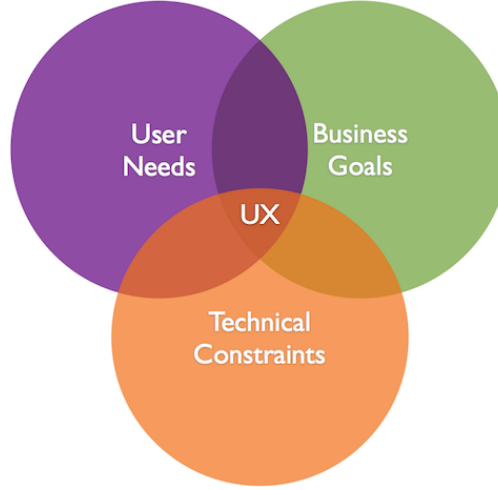


Figure 2: The three essential elements of the UX design [22].

provides access to their results to front-end part. Two other team members researched the needs of the managers, who were the end-users of the tool, following traditional user-centered design [21] procedure and developed graphical user interface of the tool, that was the web-based front-end part, according to the user research. Our supervisor prepared the simulations core of the tool by deriving only the needed for the carrier aggregation purposed simulations parts of the RAN simulator.

In case of Carrier Aggregation Tool my users were interaction designers. At the same time, I was providing their needs to our supervisor by projecting them as a set of parameters for simulations. In this type of interaction I performed a mediator role delivering knowledge and needs from one side to another and backwards. I as “mediator person” was needed for this prototype, since the interaction designers didn’t had any knowledge about technical constraints of the back-end part and simulations, that are one of the three essential elements of the UX design [22, 23] (as shown on Figure 2).

At first, after user research and discussion with our supervisor, the following simulation input parameters were selected for the tool:

- title of the particular project or case to be simulated — needed only for the graphical interface to represent and to differentiate the simulation from others in a human-readable way and has no effect on simulation

itself;

- city environment to be modeled for simulation — one of the three possible values: dense urban, urban and suburban;
- inter-site distance — the distance between two closest cellular network sites for the particular case to be simulated;
- list of radio frequency bands that should be simulated.

The choice of possible city environment types was dictated by the needs of the tool's end users. However, on the simulation core side these types are converted into the set of parameters that describe city model in details (for instance, buildings, their heights and materials, etc.). This simplification allowed to facilitate graphical user interface of the tool and provide convenient way of setting up of the city model for the simulation. At the same time inter-site distance was chosen to be out of the city model description and to be specified by user explicitly, because it is a very important parameter of the simulation and most of the times it is not dependent on the city type. Figure 3 shows the main screen of the Carrier Aggregation Tool, where user can input all the parameters.

The results of the simulations are two two-dimensional graphs that should be presented to the end users. Each graph contains a number of curves that depend on the initially chosen list of radio frequency bands. Therefore, each curve is passed from the back-end to the front-end as a set of points. The example of one of these graphs can be seen on the Figure 4.

For the front-end part of the tool it was very important to present information about already submitted simulations to the end user. Therefore, back-end part, beside graphs data, also can provide status of the simulation and date and time when it was submitted. The status of the simulation allows user to check whether it is still simulating, failed for some reasons or already finished and its results can be requested.

Simulation statuses were a very important feature for the end users of the tool as well as for the interaction designers, since they were providing a necessary feedback on the simulations. At the same time, simulation statuses provided enough knowledge about simulations for interaction designers to develop convenient and easy-to-use front-end part of the tool, even though simulations engine allowed to provide more feedback on simulation, such as intermediate results and logging information telling

BETA

Carrier Aggregation

New

Open

New project details

Title name

Test Project

Great!

Configuration parameters

Step 1: Select the environment

Dense Urban

Urban

Suburban

Step 2: Define the intersite distance (ISD) between 0 and 1000 meters

Metres

600

Step 3: Bands

	Frequency	Bandwidth	
1	800	10 MHz	
2	2600	20 MHz	

Step 4: Solution Space

	800/10	2600/20	
1	✓		
2		✓	
3	✓	✓	

SUBMIT

© 2016 Carrier Aggregation Tool

Bogdan Timus | Cristian Grama | Charlotte Kortbeek | Denis Laure

Figure 3: Main screen of the Carrier Aggregation Tool.

14



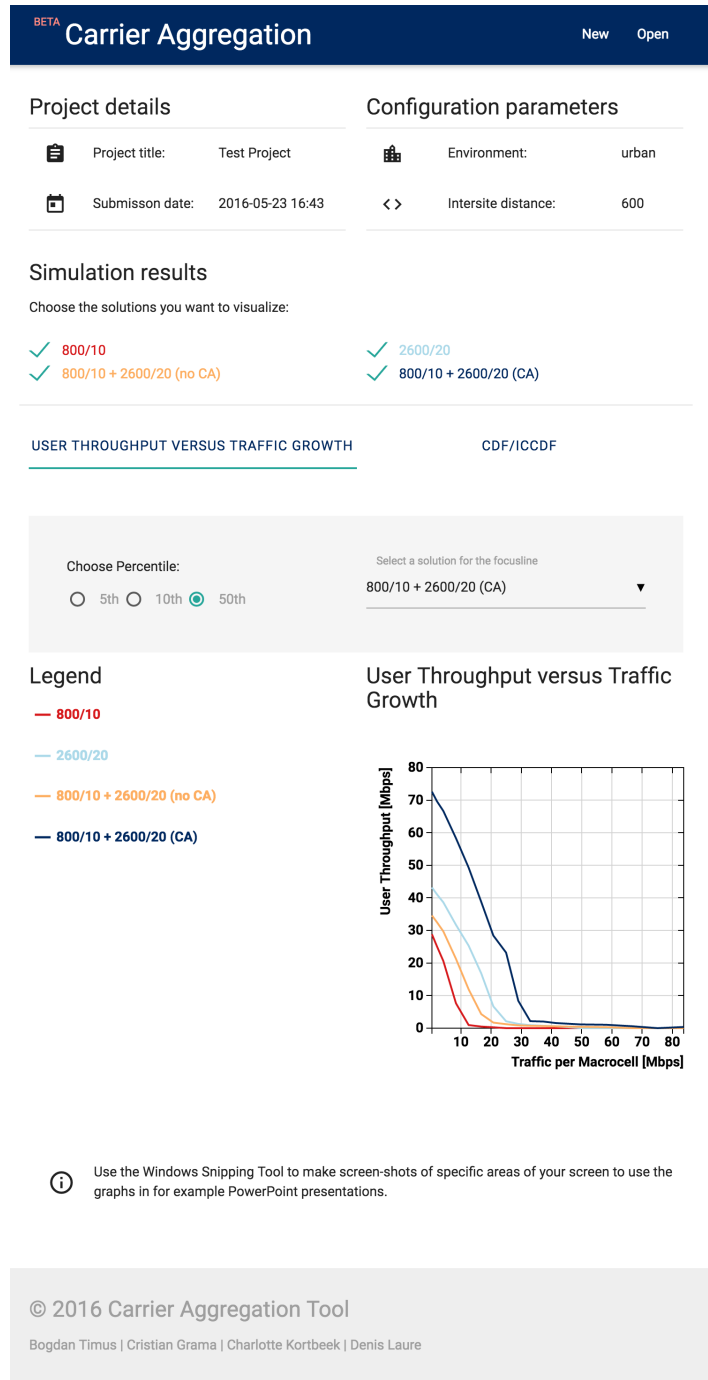


Figure 4: Results screen of the Carrier Aggregation Tool.

what part of the simulation is currently being executed. Having more detailed feedback information would unnecessarily complicate work of interaction designers as well as the design of the tool itself. Moreover, it would not provide any value to the end user.

#### 4.2.1 Design Challenges

After several iterations on the tool prototype the following changes were made into initial design.

First of all, the initial design of the graphical interface included showing information about the estimated time that the simulation should take. This feature was a result of designers' hypotheses about simulations. However, after exploring the simulation material it turned out that the RAN simulator cannot predict simulation time. Moreover, further material exploration showed that simulations for carrier aggregation purposes are relatively simple and take less than 5 minutes to proceed. Therefore, it was impossible and meaningless to show the estimated time of the simulation to the end user and the initial design of the front-end was adapted according to this.

Second, interaction designers were assuming that there should be also a "queued" status of the simulation. It would show to the end-users that their simulations are submitted but have not started yet and wait for the queue. However, the back-end launches simulations concurrently as separate computational processes, so there were no need for queuing simulations. Therefore, "queued" status was removed from initial design.

Finally, in the beginning the back-end returned about 3000 points for each curve of the graph as results of the simulation. However, such amount of points was too much to show on the tool's front-end side, due to the relatively small size of the graph area. Moreover, such amount of points complicated the work with them on the front-end part. Therefore, designers asked to reduce the number of points returned by back-end. Such a change would also improve the technical aspects of the tool, since a big number of graph points increased the size of the response body and, therefore, increased the waiting time of the response from the back-end. Therefore, the number of points for each curve was reduced to 100, as it would not impact on the graph quality and accuracy of the data, but would improve the speed of the tool and ease development of the front-end part.

The first two changes clearly show that interaction designers build their

process based only on user needs. They do not take into account the simulation material and come up with some hypotheses regarding it that sometimes are not confirmed by reality. These hypotheses result into design that just cannot be implemented because of technical constraints. Therefore, the design of the simulation-based tools should be affected by simulations material and interaction designers should be provided with the knowledge about it.

However, interaction designers do not need to know all the small details of the simulations. It would complicate the design process with unnecessary data, moving the focus from users to material and increasing the time of the design and development. Instead, they should be provided with only important parts of knowledge about material that might have effect on the design. Moreover, these parts should be formed into a format that is easy for them to understand, so that they can easily include this knowledge into design process.

#### 4.2.2 Technical Challenges

One more assumption that happened to be false was made by me and was not seen from outside of the back-end part. I was assuming that the computer that will host the back-end part will have MATLAB installed on it. Therefore, initially the simulation core was presented as MATLAB script file that was launched directly from back-end with the use of MatConsoleCtl library<sup>2</sup> that allows to control MATLAB directly from Java code. Therefore, it allowed to launch MATLAB from the back-end, save parameters retrieved from front-end into MATLAB's workspace, start simulation script with saved parameters and read results from MATLAB's workspace when simulation is finished.

However, it turned out that the chances of back-end hosting computer having a MATLAB installed were very low. Therefore, it was decided to change the approach to launching simulations. For this purpose, the MATLAB script was compiled into *.exe* executable that was launched from the back-end. This executable has two arguments: name of the file with simulation input parameters and name of the file where simulation results

---

<sup>2</sup><https://github.com/diffplug/matconsolectl>

should be saved. Both files are *.mat* files that are used by MATLAB to store the data. Therefore, the MatFileRW<sup>3</sup> library that allows to read and write data from Java into *.mat* files with proper format was used. This library allowed to save simulation parameters retrieved from front-end client in JSON<sup>4</sup> format into *.mat* file and, therefore, leave back-end API of the tool unchanged. This, in fact, allowed to leave the front-end part without any changes even with the changed approach to launching the simulations.

The challenge shows a vivid example of the knowledge about the material that is very important from technical perspective, but, at the same time, should be hidden from interaction designers. The reason for that is that the knowledge will not bring to them any value. The way simulations are launched on the back-end does not effect neither the input parameters of the simulation or their format, neither the simulations results or their format. Moreover, it is not seen nor to the interaction designers, who work on the front-end, nor to the end users. Therefore, this detail about the simulation material does not effect the design of the tool. Providing this detail to interaction designers would increase the amount of information they are working with (even though, this portion of the information is useless for them) and therefore increase the design time.

The final management of simulations launches for the Carrier Aggregation Tool looks as follows. If the back-end succeeds to find MATLAB installed on the computer, the simulations are launched directly from the MATLAB, otherwise they are launched using the executable. This choice is only seen on the back-end part and has no effect on the front-end and the graphical user interface of the tool. Therefore, no feedback or knowledge about this choice is provided to the front-end part.

### 4.3 Raw RAN Simulations Tool

The second tool that was developed for this project was the Raw RAN Simulations Tool that was requested by our supervisor. He wanted to be able to launch any types of RAN simulations remotely whenever it was needed to avoid launching them on his own computer and occupying its resources and computational power, since simulations might take quite a long time. Moreover, he wanted to be able to launch them from MATLAB

---

<sup>3</sup><https://github.com/diffplug/matfilerw>

<sup>4</sup><http://www.json.org>

on his computer, because it was one of the main environments he was working in. Therefore, no graphical user interface was needed for this tool. In this prototype the supervisor was the user of my work as well as the end user of the tool itself. At the same time, I took a role of interaction designer designing the tool itself, its behavior and interface.

Based on the Carrier Aggregation Tool development experience it was decided from the very beginning that the core of the simulations should be the RAN simulator compiled into *.exe* executable. It should also take two arguments when launched that are names of *.mat* files with simulation input parameters and simulation results.

Since both simulation core and MATLAB on the user side used *.mat* file format, there was no need to use JSON format to pass data between user and back-end. Moreover, unlike the Carrier Aggregation Tool, the Raw RAN Simulations tool can take a big number of parameters as its input and would make JSON request too big and complicated to use. Therefore, it was decided that simulation input parameters should be uploaded as *.mat* file directly to the back-end and the workflow of the tool was designed as follows.

1. User uploads *.mat* file with simulation parameters to back-end and receives unique identifier of the simulation as a response.
2. Back-end starts simulation with retrieved parameters.
3. User can check the status of the simulation by sending requests to the back-end with the identifier of the simulation retrieved on the first step.
4. Once the simulation is finished user can download its results as *.mat* file using the identifier.

Due to this workflow, the back-end configuration was changed by allowing to upload files with up to 128 MB size, while default configuration allowed to upload files with maximum only 128 kB due to security reasons. This change shows an example of how the simulation material effects the design and technical peculiarities of the tool.

One more request from the supervisor regarding the tool was that it has to be secure, meaning that it should be used only by certain persons who got access to it. Therefore, the Raw RAN Simulations Tool was secured with Basic HTTP Authentication [24] providing access to it for only registered on the back-end users with one of the following roles: *ADMIN* or *RAN*. This

type of authentication was enough even without using SSL protocol, since the tool was supposed to be used only inside Ericsson’s Intranet.

The last problem that needed to be solved for this tool was: how to upload files to back-end from MATLAB directly. Even though MATLAB allows sending HTTP requests, it supports only *application/x-www-form-urlencoded* request content type, while file uploading requires *multipart/form-data* type [25]. Therefore, external MATLAB function *urlreadpost*<sup>5</sup> was used to send simulation input files to back-end. However, this function has one big disadvantage — it does not support Basic HTTP Authentication. Therefore, it was enhanced by me to support this type of authentication and its parameters were extended by adding username and password parameters in addition to standard URL and file parameters.

This last challenge shows how the initial ideas and choices may reflect the final design. In case of the Raw RAN Simulations Tool, the choice of not using JSON format in passing simulation parameters as well as the security aspects, reflected the way users interact with the tool. The choice also reflected the development of the tool by resulting into a need to develop the MATLAB function that would provide the user with a convenient way of launching simulations. However, if this initial choice about the JSON format would be different, the final interaction aspect for the user would still remain the same — the user would still need to launch a function from the MATLAB to start the simulation. The difference would be in the technical implementation, the function would do extra work by converting parameters into the JSON format and sending them to the back-end, where they would be converted again and saved into the *.mat* file. Such a workflow would unnecessarily complicate the technical implementation and, therefore, negatively effect the reliability, speed and robustness of the whole system.

## 4.4 Administration Tool

The last tool that was developed in this project was also requested by our supervisor. He needed a tool that would allow him or other person to

---

<sup>5</sup><http://www.mathworks.com/matlabcentral/fileexchange/27189-urlreadpost-url-post-method-with-binary-file-uploading>

administrate the back-end. Requested administration functionalities included:

- viewing usage statistics for all types of simulations;
- viewing all the data related to all types of simulations;
- managing tools' users, i.e. creating new users, removing old users and setting up their roles that determine what type of simulations particular user can launch;
- managing back-end settings.

Unlike the Raw RAN Simulations Tool, the Administration Tool provides graphical user interface to allow better and more convenient data representation and management. Moreover, since it was assumed that there would be only one administrator — our supervisor — all the design was fully guided by him reflecting to all of his needs and wishes. In this prototype I again took a roles of the interaction designer and the developer.

A graphical interface of the tool was decided to implement directly on the back-end part, because this solution allowed to avoid passing raw data between front-end and back-end increasing the speed of the tool. The Main page of the Administrative Tool is shown on Figure 5.

The graphical user interface was implemented with the use of Freemarker<sup>6</sup> template engine. It allowed to easily generate HTML pages with embedded data from back-end. Additionally, Bootstrap<sup>7</sup> framework was used for the tool that allowed to simply create good looking and responsive design without spending much time on it.

The GUI design of the Administration Tool was done in a simple manner. The top header menu allows user to navigate through different pages that represent different functionalities of the tool. The footer that shows the current version of the back-end was added to easily check whether the latest version of the back-end is currently launched on the remote computer.

The Main page of the Administrative Tool represents a quick overview of all simulations with brief statistics. Initially it contained only information about the amount of simulations per status for each type of the simulation. Later, a few discussions with supervisor revealed the need to

---

<sup>6</sup><http://freemarker.org>

<sup>7</sup><http://getbootstrap.com>

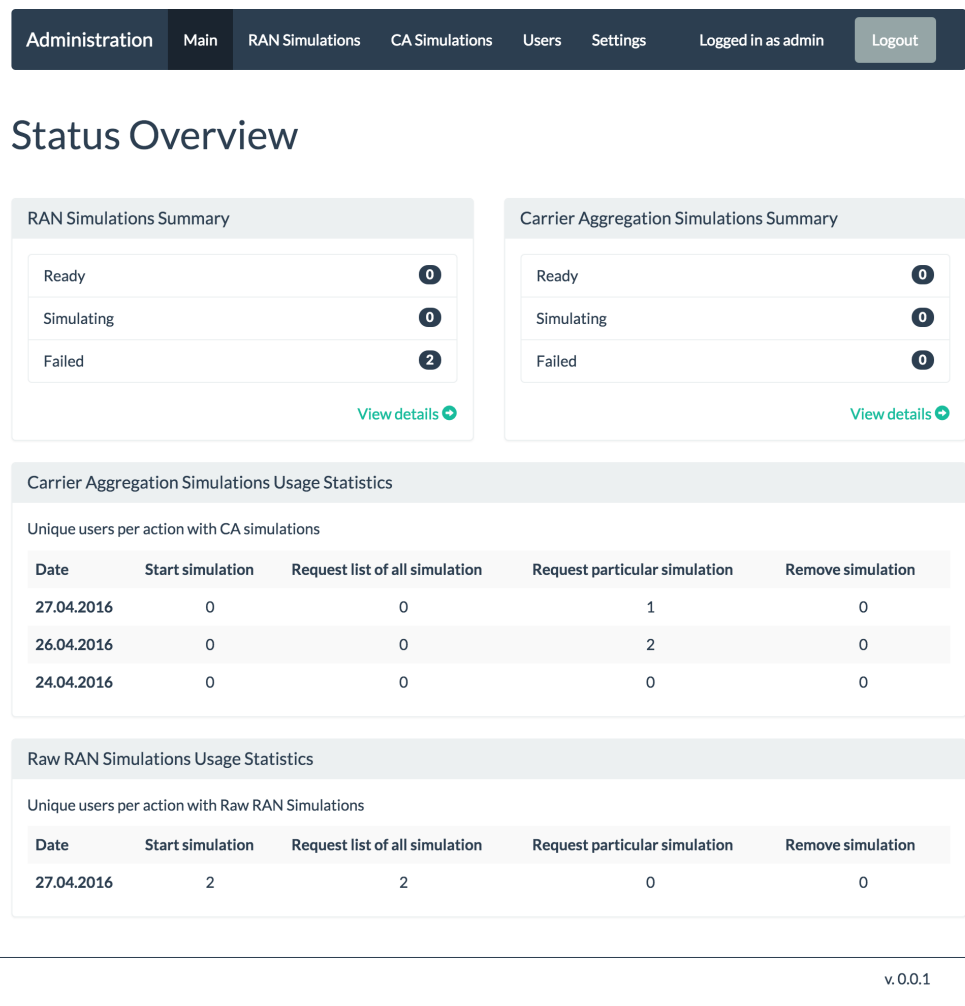


Figure 5: Main page of the Administration Tool.



see usage statistics for each type of the request of each type of simulations per unique user. In the future this feature will allow to analyze “popularity” of particular service to understand how these services can be improved. Therefore, additional blocks with usage statistics were added to the main page.

The Raw RAN Simulations and Carrier Aggregation Simulations pages provide a list of corresponding simulations with short information about them: unique identifier, status of the simulation, date when it was submitted and name of the user who submitted it. These pages also allow to quickly download results and log files for each simulation or remove it from back-end including all the related files.

The User Management page allows to quickly overview all of the back-end users and their roles, remove chosen users or create new one with ability to set up its role.

Later, additional functionality that would allow to change the back-end settings from administrative interface were requested. The first function was automatic removal of old simulations and all related files. It was needed, because results of simulations take significant amount of disk space, however, results of few weeks old simulations were unnecessary. Therefore, the Settings page was introduced to the administrative interface where the administrator is able to select for how long he or she wants to store simulations info on the back-end.

The second function that was requested was the possibility to set up folders where simulation files should be stored. It was needed to avoid saving all the files in one place that would complicate manual search of needed files in case of big number of simulations. Initially, this function was supposed to be added to the Settings page as well. However, it turned out that the simulations files should be saved on another physical drive than the one where the back-end executable was placed. Implementing such a feature in the GUI of the Administrative Tool meant that the Settings page should provide a fully functional web explorer that would provide access to the file system of the remote computer. Such explorer would be too complicated and time consuming to implement. Therefore, a compromise was found: simulation files storage folders are specified in back-end configuration files as file path strings. This approach imposes the restriction of changing the folders only on back-end restart. However, this solution was sufficient, since this functionality was not supposed to be used often.

Both added functions are examples of how the simulation material

guided and dictated the design of the Administration Tool. Even though the tool does not provide the possibility to launch the simulations, it was still dependent on technical constraints and peculiarities of the simulations.

The security aspect of the Administrative Tool was very important. Therefore, opening any page of the Administrative Tool redirects to Login page, where a person is asked to input one's username and password. If the person logged in as a user with *ADMIN* role, then access to all functionality of Administrative Tool is granted and one can freely navigate through it. Otherwise, a message is shown saying that one does not have rights to use the tool.

## 5 Discussion

One of the outcomes of the research by Brad Myers et al. [8] and Fatih Kursat Ozenc et al. [9] was that designers usually do not take into account all the details and restrictions of the implementation of the design they propose. This fact is well confirmed by my work on Carrier Aggregation Tool prototype. It showed that designers perceived simulations as a “black box” that they pass some parameters to and get the results from. They design the GUI with only user’s needs in minds without paying attention to simulations and the back-end and relying only on their initial understanding about the *immaterial material*. However, this knowledge should also be taken into account when designing simulation-based tools, because the design might be affected by restrictions and peculiarities of the material.

One of the changes in Carrier Aggregation Tool prototype showed another interesting finding about interaction designers working on simulation-based tools. It was a request from them to reduce the number of graph points in back-end response. The designers for this request were only considering design needs and did not think about how possibly the request may affect the simulation material or technical details. However, the effects might be both positive or negative. For example, some changes might affect the results of the simulations making them unreliable, inaccurate, not enough detailed or even wrong. They can also affect the simulation complexity and, therefore, the simulation time.

In this particular case, the request for reducing the number of graph points had no negative effects and even improved performance of the tool. However, in the case of the requested change having negative impact on simulation material, the team should consider solving the problem.

All of these findings show that material significantly impacts the design of simulation-based tools. Moreover, the impact is bidirectional, so the design can also affect the material by designers asking for some additional features or changes in simulations. Therefore, people designing simulation-based tools should be provided with knowledge about simulations and underlying technologies. On the other hand, the experience gained from prototyping the Carrier Aggregation Tool showed that this knowledge can be reduced to a minimum and should be provided in an easy to understand way. In this case designers would receive the necessary knowledge about the material and complete the design taking it into

account. However, the design would still be guided by user research and, therefore, will be aimed to fulfill users' needs.

As design and material impact each other, the team that works on development of simulation-based tools needs a person (that I here call “mediator person”) who will take a responsibility of delivering knowledge about simulation material to designers. This role cannot be passed to one of the designers, since it requires broad and deep technical knowledge to work, understand and *converse* with simulation material. Designers usually don't have enough technical knowledge to take the role and, therefore, it will take the designer indefensibly significant amount of time to get the knowledge about simulations. However, a “mediator person” should not only have significant technical knowledge, but also knowledge about interaction design to be able to converse with designers, determine and pass only needed for them information about simulations.

According to said above, the “mediator person” can be seen as a *boundary object* between interaction designers and simulation material, between design and simulation developers and between designers and developers. As the *boundary object*, this person allows to overcome the knowledge boundary between two parties and establish a common understanding between them [26]. Moreover, according to Carlile [27], the “mediator person” is an “effective” *boundary object*, since it satisfies all the three characteristics that make *boundary object* useful to solve a problem. Each characteristic represents a certain approach to cross the knowledge boundary: syntactic, semantic or pragmatic. The syntactic approach deals with the lack of common syntax or language between two parties. The “mediator person” discards this lack by the fact that this person provides the knowledge between two parties in a form that is common for the party that receives it. This fact also proves that the “mediator person” ensures that both parties have same interpretations within the established language, which is the semantic approach to the knowledge boundaries. Lastly, the pragmatic approach “facilitates a process where individuals can jointly transform their knowledge”. This, in fact, one of the aims of the “mediator person” — to provide interaction designers with details and constrains of the simulations and, therefore, change their initial knowledge about the material.

The “mediator” does not necessary have to be a person. For example, Sundström et al. [10] describe *inspirational bits* as a way to provide knowledge about the material to the interaction designers. In this case, *inspirational*

*bits* themselves are the “mediator”. Since *inspirational bits* need a person to create them, one may assume that this person is a “mediator person” and *inspirational bits* are just a way or form of delivering knowledge about material. However, it is not right. *Inspirational bits* provide more than just a selected direct knowledge about the material. They provide a way for the interaction designers to explore the material by themselves. Therefore, with *inspirational bits*, unlike with “mediator person”, designers need to filter important knowledge from unnecessary one by themselves. On the other hand, *inspirational bits* provide more flexibility and options to the design process as well as bring more inspiration to the designers.

The choice between *inspirational bits* and “mediator person” depends on particular project. If the project has a well defined aim and restricted amount of time (as our team had in case of our project), then it is better to have a “mediator person”. This person would allow to save time on the development and design, taking the responsibility of exploring the material from the interaction designers. Otherwise, if the project requires creativity and does not have a clear goal, *inspirational bits* can bring a needed creativity and propose more options for the design by allowing all of the members of the team to explore the material together.

In this project I was the “mediator person”. Even though I didn’t have any knowledge on simulations before starting this project, I was able to quickly dig into the material due to my technical background and experience. Moreover, since I had knowledge about interaction design and had an experience of working with designers from the team even before the project, I was able to provide them with necessary minimum of knowledge about it in a convenient way.

The prototype of the Administration Tool rises other important understandings about simulations that need to be delivered to interaction designers and that were not covered in this work. The most important of them is the size of simulation files. The RAN simulator produces files with intermediate results to store intermediate data needed during the simulation processes. At the same time, the RAN simulator stores its final results in files. All of these files take significant amount of the disk space due to the large size of these files. Therefore, keeping all of them may result into running out of disk space. On the other hand, keeping these files allows to significantly decrease simulation time, as the RAN simulator can skip some parts of its work by using previously simulated data.

Since designers perceive simulations as a “black box”, they are only

interested in keeping all of the simulation files, because it will positively reflect the design and user experience. Therefore, there is a need to provide them with understanding of the fact that always keeping all of the simulation files may result into tool stop working because of having no disk space to perform new simulation. This dilemma need to be solved and the best solution will come only with collaborative work of designers and developers, since in this case the dilemma will be examined from different edges and different approaches from different fields may be applied (or even combined together) to solve it. However, the “mediator person” will also play a significant role in this case, since he or she can guide and plan this collaborative work due to the knowledge of both sides of the team.

Summarizing, having a “mediator person” in a team working on the simulation-based tools provides a variety of benefits, such as:

- saving the time that is needed for the project by taking the responsibility of exploring the material and providing the interaction designers only with the needed knowledge about it;
- designing better solutions for the problems that arise during the project due to possible incompatibilities between design decisions and technical constraints;
- developing narrowed and, therefore, fast, reliable and robust implementations of the systems within the project by delivering designers’ needs to developers in a convenient way.

On the other hand, having no “mediator person” in a team may result into interaction designers exploring the material by themselves. Therefore, they will get more knowledge about the material and more freedom in the design process, bringing more creativity to the project.

Lastly, one more interesting moment may be considered as a part of further research. It arose after the meeting with the Sales Support department employees, where our team presented the prototypes. Even though our presentation covered even architectural aspects of our tools, it was still difficult for people to understand the fact that simulations are run on a different computer than the one we were showing graphical user interface. Therefore, further research might also consider investigation of how end users perceive simulation-based tools, what kind of peculiarities about this type of software should be provided and how this knowledge can be delivered to them.

## 6 Conclusions

This research shows that the work on the development of easy-to-use and convenient simulation-based tools should be done by combining user-centered approach with knowledge about simulations. However, interaction designers do not have enough knowledge about simulations, so they cannot take into account all their specifics, restrictions and constraints, treating them as something that can do anything and produce anything they need. Moreover, developers of simulators do not have enough knowledge about users and usually even underestimate the importance of the user research, using only their own assumptions about users for the development, that are often different from reality. Therefore, the team that works on simulation-based tools should include both interaction designers and simulation developers in order to be successful.

Moreover, the team also needs a person who would play a mediator role. This person would present needs and requests of the designers to the developers in order to adjust simulations. At the same time, this person would provide feedback and knowledge about simulations to the designers, filtering it to only needed by them and forming it into understandable for the designers format.

The results of the research also show that one of the most important aspects of simulation material for the interaction designer is the feedback on the simulations. The feedback can be represented simply with the statuses of the simulations (for example, whether it is running, finished or failed), but it is better if it can provide some additional information, for instance, estimated time of the simulations. However, the feedback should not be excessive and should not contain information that has no value for the user. The example of such excessive information is intermediate results of the simulations.

Moreover, simulation material contains some technical details that interaction designers do not take into account in their work, because of these details being hidden and not visible to them. The example of such details is the size of the files produced by the simulations. These files are stored on the back-end and, therefore, not seen to the team members, who work on the front-end part. However, such technical details are crucial for the material in general and for the simulation-based tool in particular. They may effect the design of the tool by imposing restrictions on it. Therefore, knowledge about these details should be provided to the interaction designers.

## 7 Acknowledgments

The author would like to thank Charlotte Kortbeek and Cristian Grama who were my colleagues during this project and classmates during the studies. Furthermore, a thank you to Cristian Bogdan and Henrik Artman from Royal Institute of Technology as well as Bogdan Timus from Ericsson for the assistance, help and constructive criticism throughout the project. I would also like to thank Mika Nieminen and Marko Nieminen from Aalto University for helping me with the thesis. Finally, a big thank you to my mother and father and all of my friends for supporting me throughout the studies.



## References

- [1] Ericsson. How we manage our business. [Online]. Available: <http://www.ericsson.com/res/thecompany/docs/com-facts/how-we-manage-our%20business.pdf>
- [2] —, “Annual Report 2015,” Tech. Rep., 2015. [Online]. Available: <http://www.ericsson.com/res/investors/docs/2015/ericsson-annual-report-2015-en.pdf>
- [3] C. Stephanidis, “User Interfaces for All - New Perspectives into Human-Computer Interaction,” *User Interfaces for All - Concepts, Methods, and Tools*, pp. 3–17, 2001.
- [4] J. O. Borchers, “A pattern approach to interaction design,” *AI & Society*, vol. 15, no. 4, pp. 359–376, 2001.
- [5] Ericsson. Ericsson small cells. [Online]. Available: <http://www.ericsson.com/ourportfolio/products/small-cells>
- [6] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Lossow, M. Sternad, R. Apelfrojd, and T. Svensson, “The role of small cells, coordinated multipoint, and massive mimo in 5g,” *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 44–51, 2014.
- [7] Ericsson. Ericsson software unlocks indoor performance for the mobile era. [Online]. Available: <http://hugin.info/1061/R/1948888/708082.pdf>
- [8] B. Myers, S. Y. Park, Y. Nakano, G. Mueller, and A. Ko, “How designers design and program interactive behaviors,” *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008*, pp. 177–184, 2008.
- [9] F. K. Ozenc, M. Kim, J. Zimmerman, S. Oney, and B. Myers, “How to support designers in getting hold of the immaterial material of software,” *Chi*, pp. 2513–2522, 2010.
- [10] P. Sundström, A. S. Taylor, and K. Grufberg, “Inspirational bits: towards a shared understanding of the digital material,” *Chi '11*, pp. 1561–1570, 2011.

- [11] Y. Fernaeus and P. Sundström, “The material move. How materials matter in interaction design research,” *Proceedings of the Designing Interactive Systems Conference on - DIS '12*, p. 486, 2012.
- [12] J. S. Belenguer and M. T. Smith, “An extension of computer engineering methods for interdisciplinary design,” *2011 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2011*, pp. 584–587, 2011.
- [13] P. Sundström and K. Höök, “Hand in Hand with the Material: Designing for Suppleness,” *CHI '10 - Proceedings of the 28th international conference on Human factors in computing systems*, pp. 463–472, 2010.
- [14] A. Vallgård and J. Redström, “Computational composites,” *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, vol. 522, p. 513, 2007.
- [15] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, “Experimental evaluation of wireless simulation assumptions,” *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, no. June, pp. 78–82, 2004.
- [16] J. Solsona Belenguer, “Immaterial materials: designing with radio,” *In Proc. of the Sixth International Conference on Tangible, Embedded and Embodied Interaction.*, 2012.
- [17] M. Beaudouin-Lafon and W. Mackay, “Prototyping tools and techniques,” *Institut National de Recherche en Informatique et en Automatique*, pp. 1017–1039, 2003.
- [18] E. Wilde and C. Pautasso, *REST: from research to practice*. Springer Science & Business Media, 2011.
- [19] G. Yuan, X. Zhang, W. Wang, and Y. Yang, “Carrier aggregation for LTE-advanced mobile communication systems,” *IEEE Communications Magazine*, vol. 48, no. 2, pp. 88–93, 2010.
- [20] K. I. Pedersen, F. Frederiksen, C. Rosa, H. Nguyen, L. G. U. Garcia, and Y. Wang, “Carrier aggregation for LTE-advanced: Functionality

and performance aspects,” *IEEE Communications Magazine*, vol. 49, no. 6, pp. 89–95, 2011.

- [21] C. Abras, D. Maloney-Krichmar, and J. Preece, “User-centered design,” *Bainbridge, W. Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications, vol. 37, no. 4, pp. 445–456, 2004.
- [22] J. Ivins. What i learned from teaching ux, part 1. [Online]. Available: <http://jessicaivins.net/blog/20130311-What-I-Learned-From-Teaching-UX-Part-1.html>
- [23] L. Vizard. 3 ingredients for a successful ux strategy. [Online]. Available: <http://blogs.adobe.com/creativecloud/3-ingredients-for-a-successful-ux-strategy/>
- [24] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, “Http authentication: Basic and digest access authentication,” 1999.
- [25] L. Masinter, “Returning values from forms: multipart/form-data,” 2015.
- [26] S. L. Star and J. R. Griesemer, “Institutional Ecology, ‘Translations’ and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, 1907-39,” *Social Studies of Science*, vol. 19, no. 3, pp. 387–420, 1989. [Online]. Available: <http://sss.sagepub.com/cgi/doi/10.1177/030631289019003001>
- [27] P. R. Carlile, “A Pragmatic View of Knowledge and Boundaries : Boundary Objects in New Product Development,” *Organization Science*, vol. 13, no. 4, pp. 442–455, 2002.
- [28] M. Curtin, “Write once, run anywhere: Why it matters,” *Technical Article*. <http://java.sun.com/features/1998/01/wo>, 1998.
- [29] J. Bishop and N. Horspool, “Cross-platform development: Software that lasts,” *Computer*, vol. 39, no. 10, pp. 26–35, 2006.
- [30] R. Johnson, J. Hoeller, A. Arendsen, T. Risberg, and D. Kopylenko, *Professional Java development with the Spring Framework*. John Wiley & Sons, 2005.

- [31] J. Arthur and S. Azadegan, “Spring framework for rapid open source J2EE web application development: A case study,” *Proceedings - Sixth Int. Conf. on Softw. Eng., Artificial Intelligence, Netw. and Parallel/Distributed Computing and First ACIS Int. Workshop on Self-Assembling Wireless Netw., SNPD/SAWN 2005*, vol. 2005, pp. 90–95, 2005.
- [32] L. DeMichiel and M. Keith, “Java persistence api,” *JSR*, vol. 220, 2006.

## A Back-end Part Description

The back-end part of the solution was developed in Java with the use of Spring Framework<sup>8</sup>. The choice of the programming language was dictated by two significantly important facts. First of all, I already had enough knowledge and experience in Java that allowed me to start working on the solution and prototyping from the very beginning without learning programming language, its specifics and common practices. Secondly, Java is a platform-neutral language that follows “Write Once, Run Anywhere” principle, which means that once compiled Java code can be then run on any popular platform [28]. Therefore, it allows to create platform-independent applications [29] that is very important in this particular case for two reasons:

- in the very beginning it was not clear whether the machine that will host a server will run Windows or Linux operating system;
- the development process was done by me in Mac OS X operating system that was not target platform for the back-end platform, however it was important for me due to my experience and practices and, therefore, allowed to decrease development time.

Spring is the most popular lightweight framework for development of Java-based enterprise applications [30]. It allows to create complicated enterprise applications using minimal amount of code [31]. Moreover, the code itself is also very simple to write, read and understand due to such features of the framework as dependency injection and ready-to-use services. This advantage was very significant for the project since it was important that the person who will take the responsibility on maintaining the back-end part can quickly start working without putting much effort into understanding of the code.

In addition, Spring Boot<sup>9</sup> framework was also used for this project. The main purpose of the framework is to simplify and automate common set ups and configurations of Spring applications. In addition, it also fastens and simplifies the process of adding extra service to the application, such as security management and database support. Moreover, it provides

---

<sup>8</sup><https://projects.spring.io/spring-framework/>

<sup>9</sup><http://projects.spring.io/spring-boot/>

embedded servlets, such as Apache Tomcat<sup>10</sup> or Jetty<sup>11</sup>, that allows to compile the application into ready-for-production Java executable that when run automatically launches the servlet and starts serving web services provided by the application. All of these advantages allowed to start prototyping and testing of the back-end part very quickly.

The back-end application code is represented with following packages:

- **configuration** — contains classes responsible for back-end configuration. These classes include security configuration, which determines services access rules, and processes configuration, which determines maximum number of simultaneous simulations that can be run.
- **process** — contains classes that launch particular simulations and manage their work, such as saving information and results.
- **model** — contains classes that model application entities. These entities represent: simulations info (status, path to related files, etc.) for different types of simulations, user information (user name, encoded password, roles), applications settings (period for how long simulation files should be stored), usage statistics (who and when used particular function of the back-end). These classes are needed to be able to automatically store, change and retrieve entities from database using JPA [32] technology without writing any database related code and queries. At the same time it allows to represent this data in JSON format and convert it to Java objects from JSON with the use of Jackson library<sup>12</sup>.
- **repository** — contains models' repository interfaces that describe methods to access models data from database. These interfaces are required by JPA.
- **web** — contains web controller classes that describe back-end's web interfaces and their behaviour.

One of the requirements to the back-end part was that it should be launched as Windows service, so that it always runs in background. For

---

<sup>10</sup><http://tomcat.apache.org>

<sup>11</sup><http://www.eclipse.org/jetty/>

<sup>12</sup><https://github.com/FasterXML/jackson>

this purpose WinSW<sup>13</sup> tool was used. It also provided some additional features that were very useful, such as automatic restart of the service and automatic saving of application logs into files.

---

<sup>13</sup><https://github.com/kohsuke/winsw>